Transient Surrogate Modeling for Thermal Management Systems

Andrew Van Zwieten,* Gokcin Cinar,† Elena Garcia,† Jonathan C. Gladin,† Dimitri N. Mavris,‡
Aerospace Systems Design Laboratory, Atlanta, GA, 30332

In typical multidisciplinary design optimization problems, with varying missions, aerodynamic data, engine data, Thermal Management Systems (TMS), and other parameters it can take upwards of millions of runs to cover the full design space which result in extremely large computational burden, reducing the effectiveness of the design process. The objective of this work is to create a technical approach, leveraging existing concepts in surrogate modeling and other relevant fields, for the creation of a Transient TMS surrogate model. This work utilizes Design of Experiments to intelligently sample a model's design space, surrogate modeling to enable instantaneous predictions, and state space modeling to allows surrogates to carry predictions forward. By leveraging these three components, a six step methodology was developed. This paper explains the developed methodology with an application on a notional Transient TMS. We first pre-process the time dependent data and possible correlations between input variables, then break down a set of input variables into a series of step functions which represent input schedules in a fraction of the time. We create Artificial Neural Network models to predict the future response of a metric of interest using the current response of both the input step functions and the corresponding output. We finally test whether the surrogates which were created with step functions could be used to predict the future response of the metric of interest for a full time trace of a sample aircraft mission. We show that this methodology yields acceptable predictions for both the partial and full time trace, with a maximum error of 5% and 10%, respectively.

I. Nomenclature

ACS	=	Air Cycle System
$ATTMO$	=	Air Force Research Laboratory Transient Thermal Modeling and Optimization
DOE	=	Design of Experiment
$EFTMS$	=	Engine Fuel Thermal Management System
RPM	=	Revolutions Per Minute
MDO	=	Multidisciplinary Design Optimization
TMS	=	Thermal Management System
u	=	Input Variables
VCS	=	Vapor Cycle System
x	=	State Variables
y	=	Response
pps	=	Pounds per Square Inch

II. Introduction

Thermal constraints need to be considered in the design process when solving Multidisciplinary Design Optimization (MDO) problems for aircraft systems. This will become increasingly important as future aircraft systems are projected to require much higher thermal demands[1]. MDO problems can take upwards of millions of runs to cover the full design space[2]. These high fidelity models reduce the effectiveness of the design process due to their extremely

*Graduate Research Assistant, School of Aerospace Engineering, AIAA Student Member
†Research Engineer II, School of Aerospace Engineering, AIAA Member
‡S.P. Langley Distinguished Regents Professor and Director of ASDL, School of Aerospace Engineering, Georgia Tech, AIAA Fellow

large computational burdens. This research is intended to reduce the computational burden by focusing on reducing the run time of current high fidelity thermal models. In addition to the reduced run time, implementation of surrogates can provide TMS transient behavior to larger aircraft level tools. Hence, the need to be able to span a large range of variables to work in unknown steady state points. The focus is on improving the development of the methodology capable of creating time domain surrogate models for TMS. The model abstractions can consider component, cycle or system level performance; it is expected that there are trade-offs between the execution speed of the surrogate model and the resolution of the thermal state data. Transient surrogate models are required to capture the multi-scale temperature response inside of the system and to handle transient response of variable boundary conditions such as heating and cooling loads and the engine operating conditions. Both steady and transient behavior must be captured, and it is not known a priori where these different states will occur.

Through machine learning techniques, data from a given model can be sampled and trained in order to model how the outputs react to different inputs into the system. Surrogate modeling techniques allow for complex models to be broken down into equations between input and outputs with the proper surrogate training[3]. Training surrogates does become more complex for systems that are time dependent. When time is introduced as a variable, an added layer of complexity is created due to time's dual nature as a variable and as part of the response. The impact of time on a system depends on whether the system is performing transient operations or is in steady state. In transient operation, a system changes as time increases. Whereas, in steady state operations, the system will remain in the same state unless a perturbation to the system is added.

The objective of this work is to leverage existing concepts in surrogate modeling and other relevant fields, for the creation of a Transient TMS (TTMS) surrogate model consistent with the objectives. The TTMS surrogate modeling approach should be able to track the temperature and flow of all independent closed-loop working fluids to ensure all constraints are met. It should be able to keep track of all power requirements, primarily when power is required to be pulled off the engine which allow for losses in thrust to be accounted for. It should also be able to provide brief estimates regarding the weight and volume of the component/system, as well as calculating the change in mass of the on-board fuel.

Two main methods for approaching the problem of creating transient surrogate models are typically used. The first method treats time as an additional variable[4][5][6][7][8]. The second method treats time as an independent variable[9][10]. There are strengths and weaknesses to both of these methods. When surrogates are trained with time as an input variable, the accuracy may by better for shorter simulations and circumstances where it is known whether the model is behaving in steady state or not. However, the relationship between time and responses may degrade the accuracy of model because some spans of operations may have higher time variance and some have lower or no time variance. When time is not used as a variable, a state space[11] modeling approach may be used where by each state is predicted by the inputs to the system and the previous states values. Since it is required to track the outputs during both transient and steady state, this method was selected because it provides more versatility.

III. Methodology

The Air Force Research Laboratory Transient Thermal Modeling and Optimization (ATTMO) toolbox[12] based in Matlab/Simulink is used to simulate the TTMS responses. This model includes fuel tanks, engine fuel thermal management system (EFTMS), vapor cycle system (VCS), and an air cycle system (ACS). Input parameters time series provide the model with inputs to the system and may be varied to simulate different missions. Different missions sets may be used for sampling data and validation of surrogate performance. Once a simulation is finished outputs of interest are saved to the MATLAB workspace for data analysis and surrogate fitting.

The methodology used for creating these surrogates is broken down into six different steps. Initial steps require establishing an understanding of how the model works, specifically what inputs are expected in order to train surrogates appropriately. This starts by sampling the initial data, analyzing possible correlations, and then parameterizing the input schedule. The following three steps focus on sampling the data and creating the surrogates. First a Design of Experiment (DoE) is created based off the parameterized inputs, then a sampling of the necessary data from the TMS model, and finally creating the surrogates and evaluating their performance. Each of these steps will be discussed further in the sections below.

A. Sample Initial Data

When determining where to sample data from the model, it is important to only consider the feasible areas of the design space. Training the surrogate with unrealistic input combinations would add a significant amount of training

Table 1 TMS Model Input Ranges

Input	Min	Max	Unit
Wf36 History	0	20	pps
Wf66 History	0	20	pps
Pt F350 History	2	60	psi
Tt F330 History	523	120	°R
Wt F330 History	17	400	pps
Avionics History	30	100	kW
DEW History	45	187.5	kW
EFTMS Loads History	79.	331.25	kW
FADEC History	12	50	kW
Fuel Chiller Flow History	1e-4	7	pps
Bleed Pressure History	860	1200	kPa
Bleed temp History	370	480	°C

time and could introduce unnecessary error to the surrogate. The inputs into the model are shown in Table 1. Each input has a given minimum and maximum value based on limits suggested by AFRL. The range between the minimum and maximum input values determines the range which the TMS model may be sampled. Apart from the ranges, a more critical analysis of the inputs was needed in order to understand the type of missions being performed and better sample the space. An input schedule was provided by AFRL to serve as an example mission profile. The input schedule was sampled to analyze trends in the inputs. As shown in Fig. 1, many of the inputs differ in amplitude but have similar profiles over time. Therefore, the next step was to analyze the possible correlations between inputs.

B. Analyze Possible Correlations

The inputs from the initial data indicate that there is correlation between the input schedules however the amplitudes differ by a large amount. To better analyze these similarities, inputs were normalized using min-max scaling. Data normalization is commonly used in data processing to standardize design variables[13] and is good practice especially if some data sets vary widely while others don't. Min-max scaling is represented in Eq. (1) where x' is the normalized data and x is the original data set. The minimum value of a data set is subtracted from each data point and then divided by the range of the values (maximum value minus minimum value). This has the result of normalizing any data set to values between 0 and 1.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

After normalization, the dependencies between inputs needed to be identified before creating a DoE. If dependencies are not accounted for, the DoE wouldn't represent the reality since DoEs assume that variables are all independent. To reveal dependencies and their strength, correlation coefficients were calculated. These correlation values are given in Table 2. Of the twelve input variables, two were unchanged throughout the mission, four were 100% correlated to each other, five were greater than 65% correlated to another variable, and one had vary little correlation to the other variables. The fewer independent input variables the less computation time is needed to sample the surrogate. Avionics, Tt_F330, and Wf66 were selected as the three independent variables and dependent variables were chosen based on their correlation to the independent variables. Avionics was selected as one of the independent variables since there were five other highly correlated variables, however any of the three variables with 100% correlation could be substituted in its place. Tt_F330 was selected as the second independent variables since there were two highly correlated variables associated with it. Finally, Wf66 was selected as the third input variable since it was not highly dependent on other variables. Bleed pressure and bleed temperature inputs were left out to save time and computational power since their inputs were constant throughout the mission. One thing that should be noted is that Pt_F350 had a slightly higher correlation to W_F330 than Avionics, however the Avionics correlation was still significant and selected to avoid double dependence. The correlations between dependent and independent variables are illustrated in Fig.2

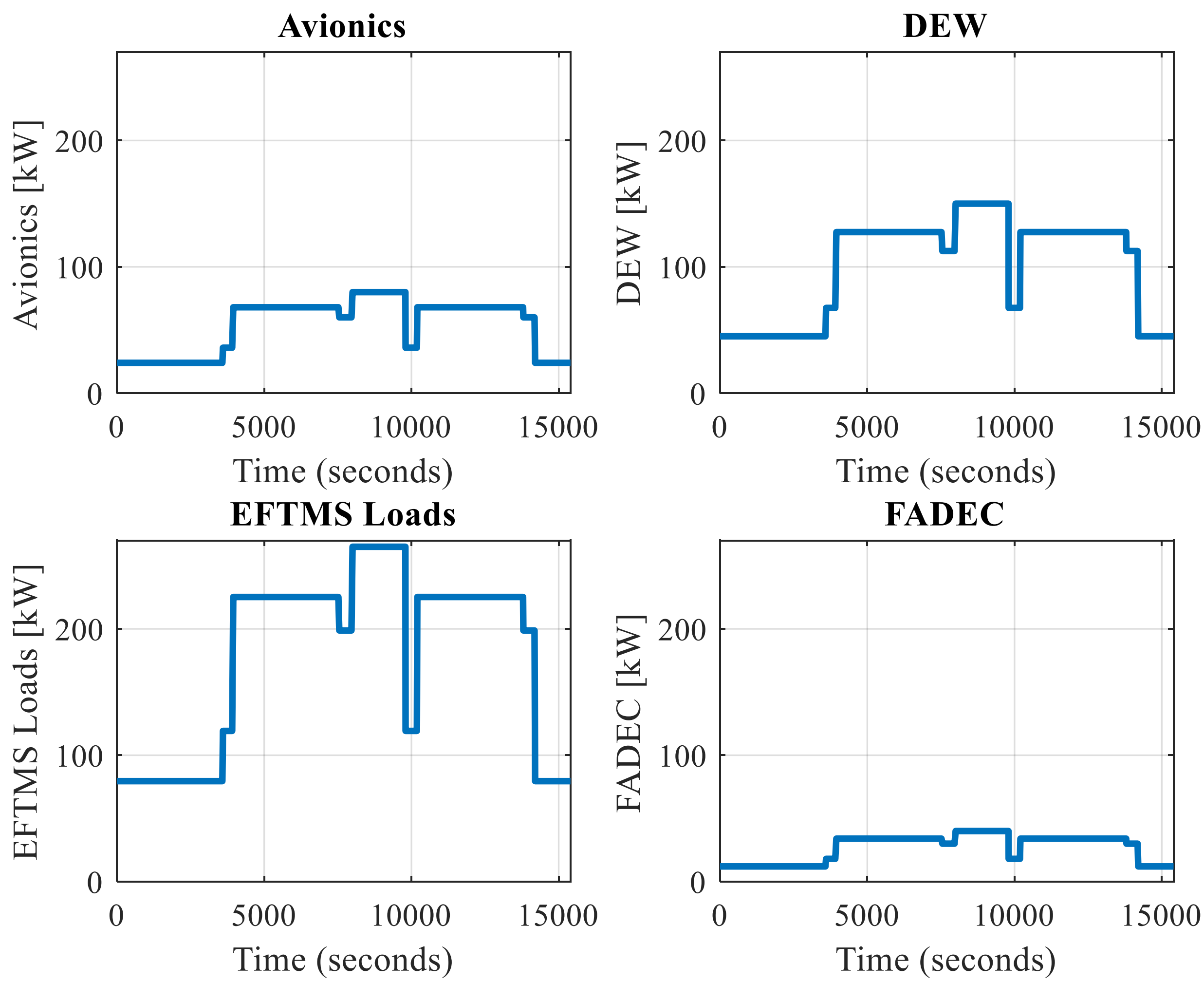

Fig. 1 Example of Input Schedules

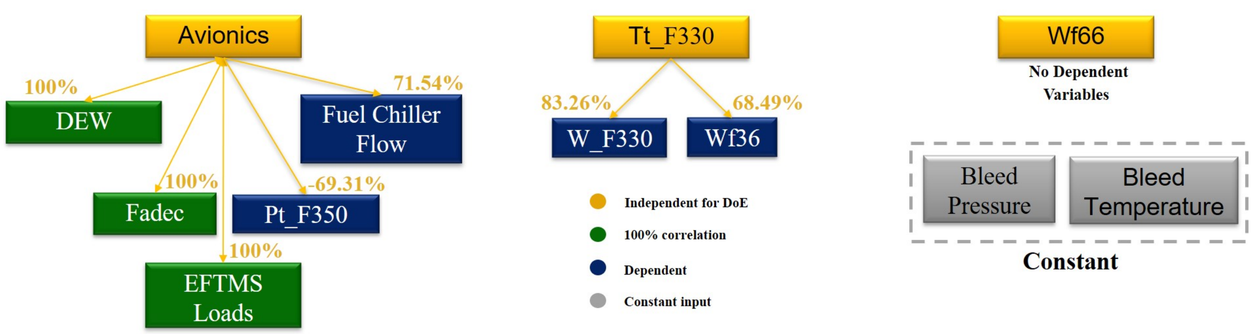

Fig. 2 Illustration of Correlation Between Input Variables

C. Parameterize Input Schedule

Once correlations were selected, the input schedule needed to be parameterized in a way such that the inputs for sampling represented the inputs to the systems during a mission. The inputs into the model when performing a sample mission largely changed instantaneously. Since most of these input change instantaneously, when looking at the full schedule, it can be illustrated as multiple step functions attached in series. This is illustrated in Fig.3, where a notional

Table 2 Input Variable Correlation Matrix

	Avionics	DEW	EFTMS	Fadec	FCF	Pt F350	Tt F330	W F330	Wf36	Wf66
Avionics	1.0000	1.0000	1.0000	1.0000	0.7154	-0.6931	0.1419	-0.2370	0.6202	-0.0738
DEW	1.0000	1.0000	1.0000	1.0000	0.7154	-0.6931	0.1419	-0.2370	0.6202	-0.0738
EFTMS	1.0000	1.0000	1.0000	1.0000	0.7154	-0.6931	0.1419	-0.2370	0.6202	-0.0738
Fadec	1.0000	1.0000	1.0000	1.0000	0.7154	-0.6931	0.1419	-0.2370	0.6202	-0.0738
FCF	0.7154	0.7154	0.7154	0.7154	1.0000	-0.6197	-0.0235	-0.3149	0.4660	-0.0325
Pt F350	-0.6931	-0.6931	-0.6931	-0.6931	-0.6197	1.0000	0.5445	0.8326	-0.1493	0.0444
Tt F330	0.1419	0.1419	0.1419	0.1419	-0.0235	0.5445	1.0000	0.8932	0.6849	0.1948
W F330	-0.2370	-0.2370	-0.2370	-0.2370	-0.3149	0.8326	0.8932	1.0000	0.3881	0.0811
Wf36	0.6202	0.6202	0.6202	0.6202	0.4660	-0.1493	0.6849	0.3881	1.0000	0.2296
Wf66	-0.0738	-0.0738	-0.0738	-0.0738	-0.0325	0.0444	0.1948	0.0811	0.2296	1.0000

mission is broken down into three different step functions. By treating the full mission as a series of these step function inputs, a simpler sampling procedure may be used. This allows for the model to be sampled by the independent inputs only being changing at one instance versus analyzing responses while inputs are constantly changing.

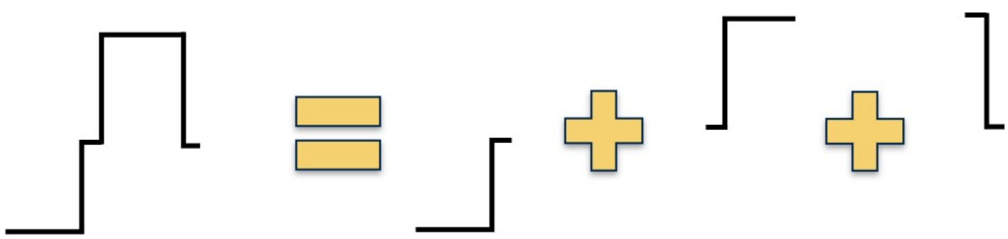

Fig. 3 Input Schedule as a Series of Step Functions

Dependent variables were parameterized to reflect their correlation to independent variables. Two variables were used for each input.The first variable was used as the initial value and the second as the final value, to parametrize the step function. Once independent variables were created, dependent variables were selected based off a uniform random number within the correlation range. An example shown for the dependent variable, Fuel Chiller Flow, is shown in Fig.4. There is a 71.5% correlation between the two inputs. Therefore, if Avionics has a normalized value of 0.5 then Fuel Chiller Flow will have a normalized value between 0.36 and 0.64.

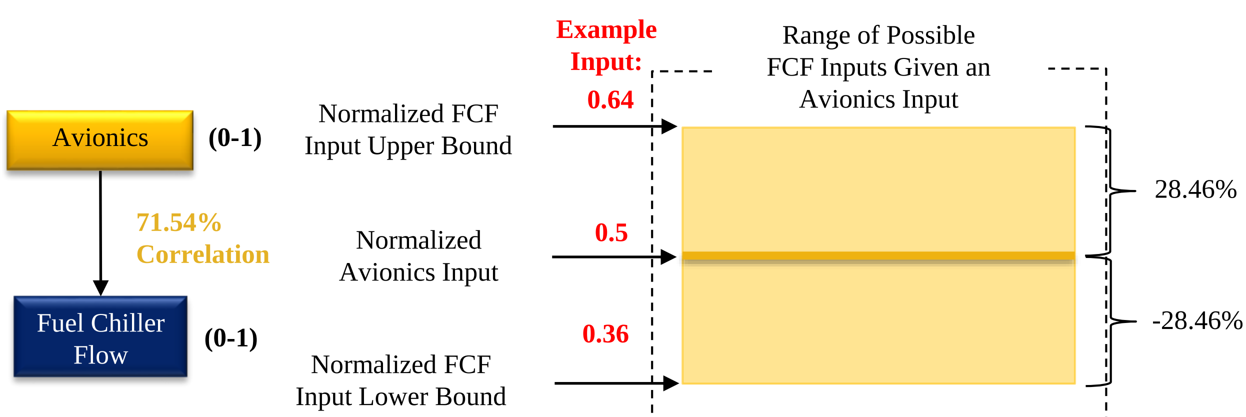

Fig. 4 Parameterization of Dependent Variables

D. Create A DoE

A Design of Experiments (DoE) is a set of experiments which are created for the purpose of gathering the maximum amount of information pertaining to a system while reducing the total number of experiments for the given system[14]. Using a DoE allows for information to be gathered from the TMS model without having to perform an excessive amount of runs. With the 3 independent input variables and 7 dependent input variables there are a total of 10 input variables. For creating inputs into the model using a step function the DoE must be created to include the initial input as well as the final input. Therefore, each input variable includes two separate variables as illustrated in Fig.6. This increases the DoE to include six independent variables and fourteen dependent variables. To create this DoE a 2 level full factorial was used for independent variables in order to cover all edges of the design spaces with one center point. This results in $2^6 + 1 = 65$ cases. Once completed, a space filling Latin hypercube of 200 points was augmented to the 2-level full factorial DoE to sample the rest of the design space. Good practice typically uses at least two times as many space filling cases as edge cases. In order to account for possible model failure or poor data an addition two hundred space filling cases were used to cover the interior sufficiently. This resulted in a total of 265 cases for the DoE. Both DoEs are illustrated in Fig.5.

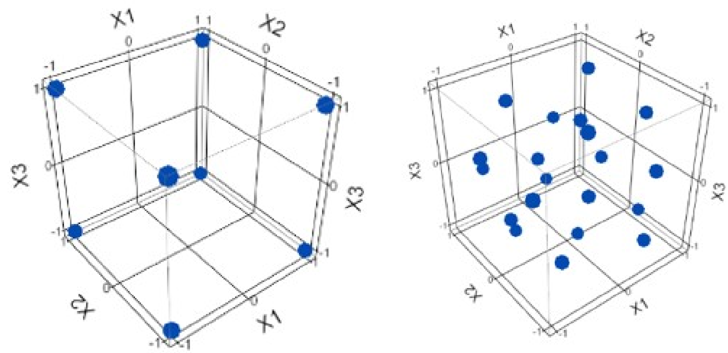

Fig. 5 DoEs: 2 Level Full Factorial and Space Filling Latin Hypercube[15]

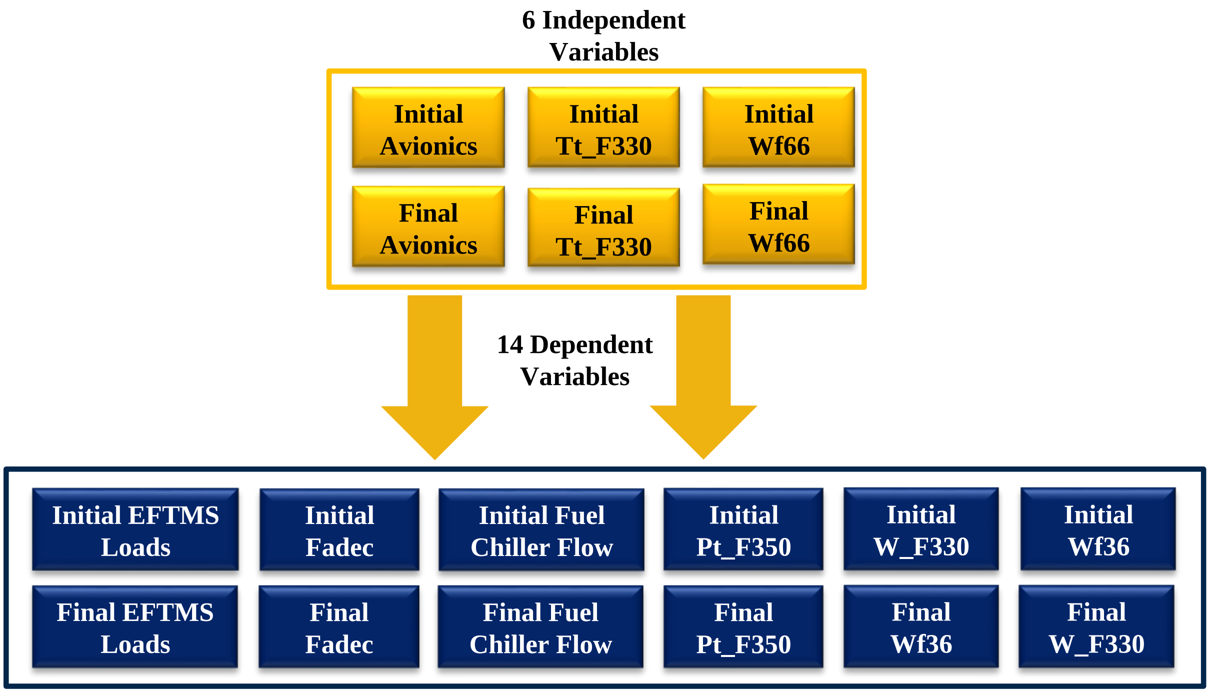

Fig. 6 Dependent and Independent DOE Variables

E. Sample DoE

Sampling of the TMS model was done using the DoE with initial and final values for inputs. Before sampling the model, the sampling rate and sampling time were needed to be determined. An appropriate sampling rate captures all transient data without being too dense as to generate an excessive amount of data. By analyzing the model responses over the most transient areas, the outputs of interest were plotted versus time to determine necessary sampling rate. The model is the most transient in the first seconds of the simulation which would require a very dense sampling rate to capture the initial responses, and thus resulting in a very large data set. Another consideration is that at the start of the simulation, response variability is driven more by the model convergence than any input variation. This is illustrated in Fig.7 where the first second of the simulation is plotted across the entire range of DEW values with minimal impact. When sampling at the start of the simulation, the model also takes longer to reach a steady state for most responses. For these reasons, the sampling procedure was selected to start after the model has was running for 50 seconds to avoid initial model convergence oscillations. The following inputs are added to the same simulation as shown in Fig.8 until the DoE is finished sampling. By starting sampling after steady state, a sparser sampling rate is possible.

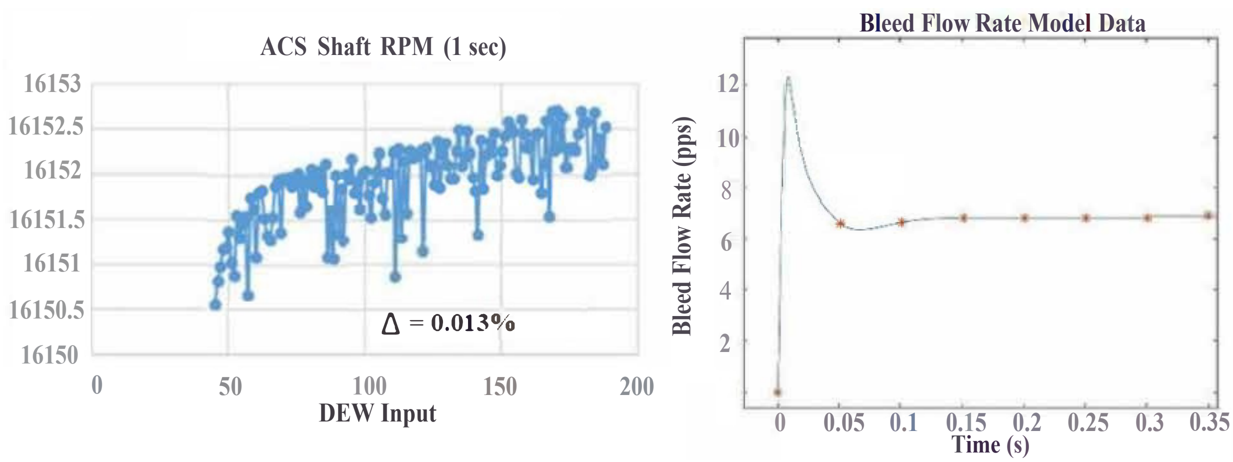

Fig. 7 Sampling Rate and Sampling Time

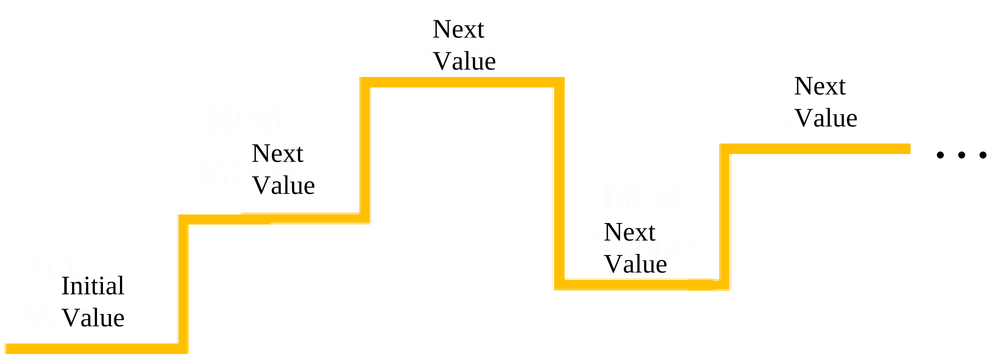

Fig. 8 DoE Inputs Schedule

Determining the sample duration is based on the transient duration of the model after input change. The data must be able to capture the transient data after the input change until the model reaches steady state. If the sampling time is too short, the full response to an input will not be captured. If the sampling time is too long, it will add unnecessary data and could skew the surrogate by sampling for too long at steady state. An example of model responses is shown in Fig. 9. The left side shows the most transient area during model response with markers every seconds. 1 second time steps appear to be plenty dense, so the sampling rate of 1 second was determined to be sufficient. The right side of the figure shows the sampling time. The time needed to reach steady state after an input change was approximately 50 seconds. For the response shown, steady state isn't quite reached yet but most of the transient area is covered.

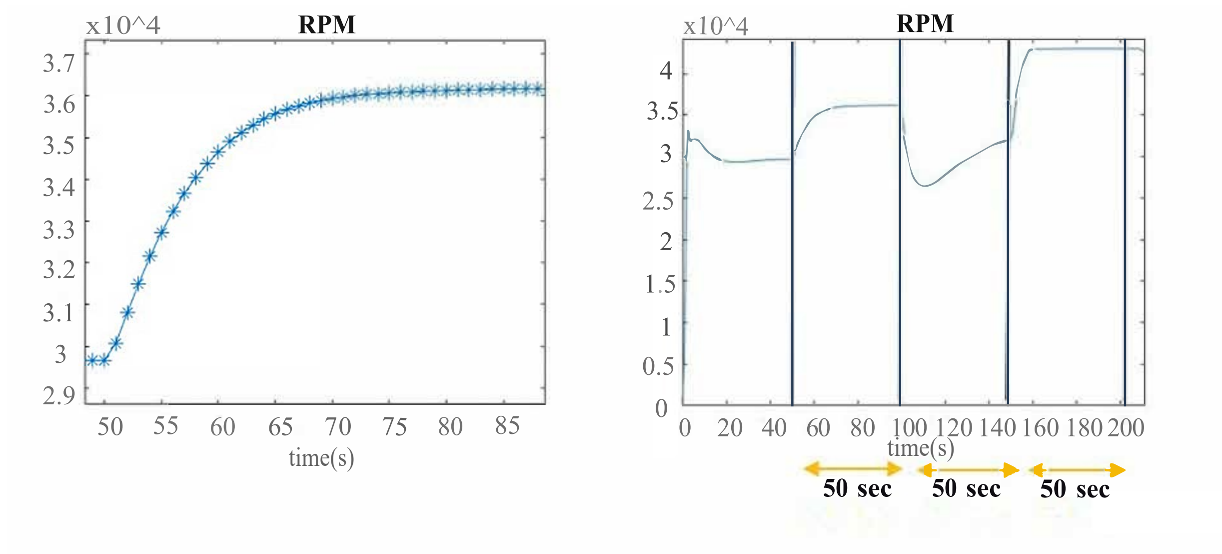

Fig. 9 Sampling After Steady State

F. Create and Evaluate Surrogates

The surrogates were created by using a discrete state space representation of the model as expressed by Eq. (2). This method takes the input vector, u_{t+1}, and previous state vector, y_t, to predict the next state y_{t+1} in the following time step. The inputs to the system and the initial starting states are known, the challenge is predicting the future states of the system. In this case, the states of interest are model responses. Once the response of the next time step is predicted, the predicted response is used to predict the following response. This process continues until the final time step as illustrated in Fig.10.

$$\overrightarrow{y_{t+1}} = f(\overrightarrow{y_t}, \overrightarrow{u_{t+1}}) \tag{2}$$

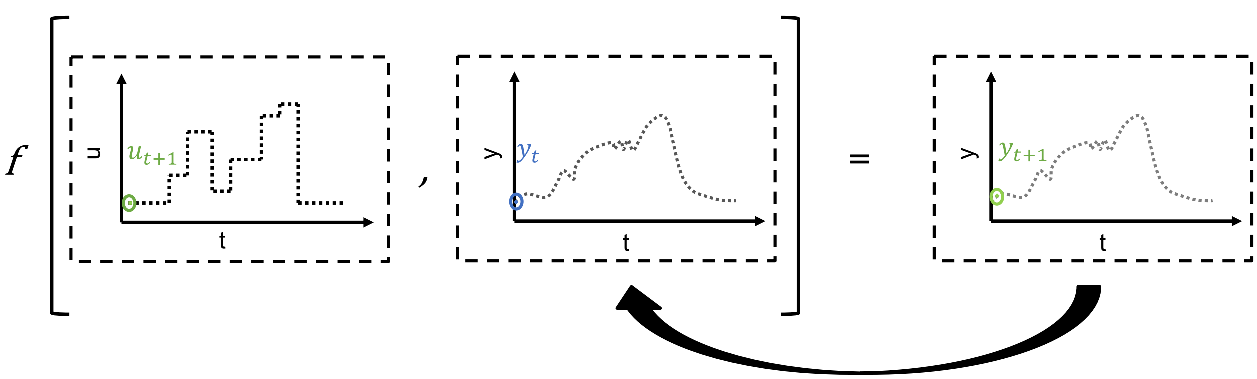

Fig. 10 Discrete State Space Surrogate Modeling Illustration

When sampling the data, the model failed to converge on 35 cases leaving 495 total cases to train and test surrogates with. Of the 495 total cases, 445 were used for training and 50 were used for testing the surrogate (10%). From the sampling procedure a time step of 1 seconds was used. The surrogate model was created with Artificial Neural Networks (ANN) using 1 or 2 hidden layers, and a varying number of nodes. Out of the 445 training cases 70% were used as training data to generate initial surrogate, 25% were used for validation to refine surrogate, and 5% were used as test cases to evaluate the surrogates. 50 additional test cases were used to determine usefulness of surrogate outside of training

The ANN models used have an input layer, output layer, and one or two hidden layers. The hidden layers contain nodes which used equations to predict the response expected from a set of inputs after training. The goal when training an ANN is to find the minimum number of feasible nodes which gives an acceptable response. The number of hidden layers and nodes in each ANN model depends on the specific problem. However, generally speaking, increasing the size of the network gradually improve the fit for the training and validation data. Too many nodes can often lead to overfitting, which means the model will work well for training and validation data, but can perform much worse for test data.

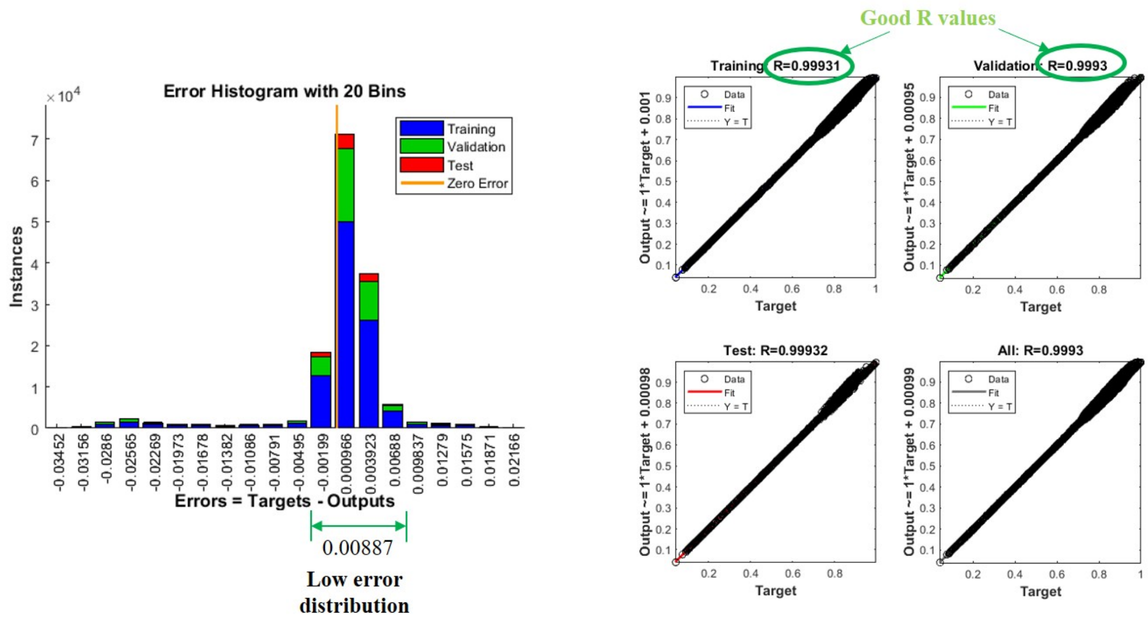

Fig. 11 Example of a Test for Goodness of Fit

There is a clear trade-off between model complexity and likelihood of an overfitting problem. More complexity also results increased computational burden which requires more memory and time. Therefore the approach used was to start with a relatively low number of nodes, around the same number as input parameters, and train the ANN model multiple times. Retraining the same model structure with same inputs and outputs can give very different fits, especially as the model complexity is increased, due to the use of different initial conditions. After training, the goodness of the fit for training, validation, and test data was checked. If the fit performed well for all data sets, a less complex model was attempted by reducing the number of nodes. Reducing the number of nodes was continued until the fit becomes unacceptable. In cases that the fit did not perform well for at least one data set, a more complex model was tried by increasing the number of nodes or layers. Nodes or layers would continue increasing in number until the desired performance was obtained.

An example of the goodness of fit test is shown in Fig.11. These graphs give the error histogram as well as the training and validation R values. The error histogram, on the left, gives the error distribution. The R values, on the right give the correlation coefficient between the model data and predicted data. The tolerance for this data depends on the specific problem. Since the surrogates are feed forward, any error produced will compound upon itself so an R value as close to 1 as possible and a low error distribution is desired.

G. Noise Detection

The physics based TTMS model often provides noisy results from certain input schedules. These oscillations occur during constant input so the results are likely due to model error and not indicative of real-world responses. An example of this noise may be seen can be seen in Fig.12. Surrogates trained on noisy data will have difficulties providing accurate predictions. It is important to find an intelligent way to deal with noisy data when creating surrogates.

A noise detection algorithm was created in order to determine which cases were noisy and which were smooth. The algorithm was created to detect the number of peaks within a given response. A response was flagged as noisy if the number of peaks were greater than 6 for one DoE case. In Fig.13, a smooth response is shown on the left which resulted in only one peak while the noisy response on the right resulted in 19 peaks. During initial sampling, there were 179 out of 236 (76%) cases which were determined to be noisy based on this criteria.

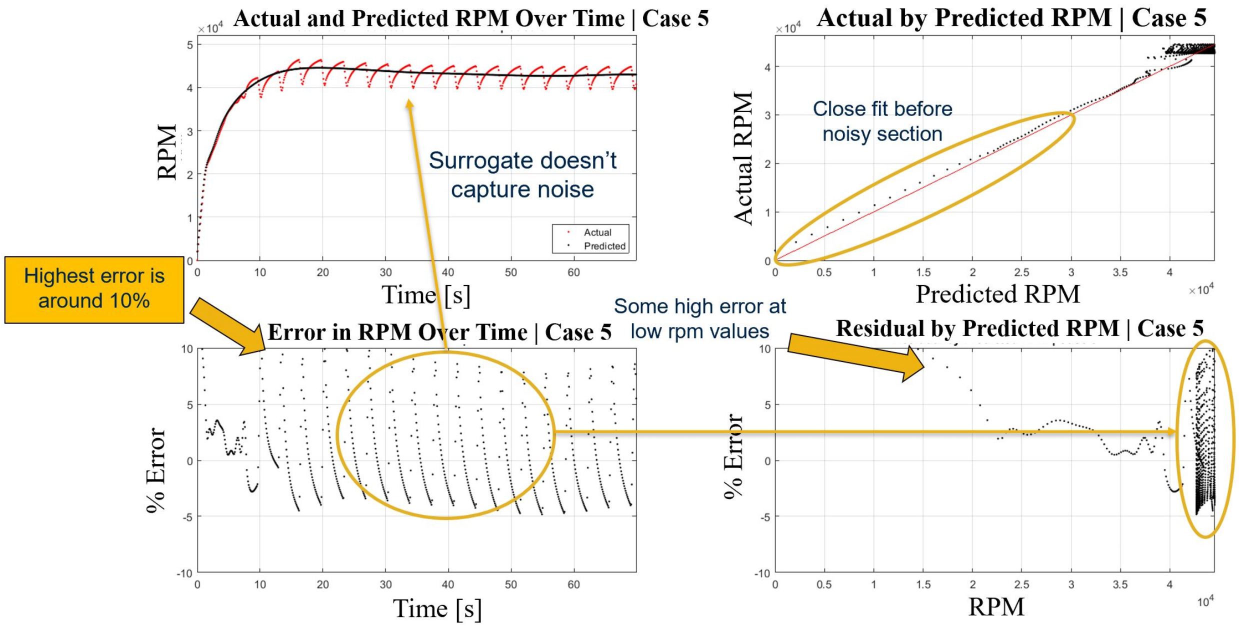

Fig. 12 Initial Training Results for RPM Surrogate

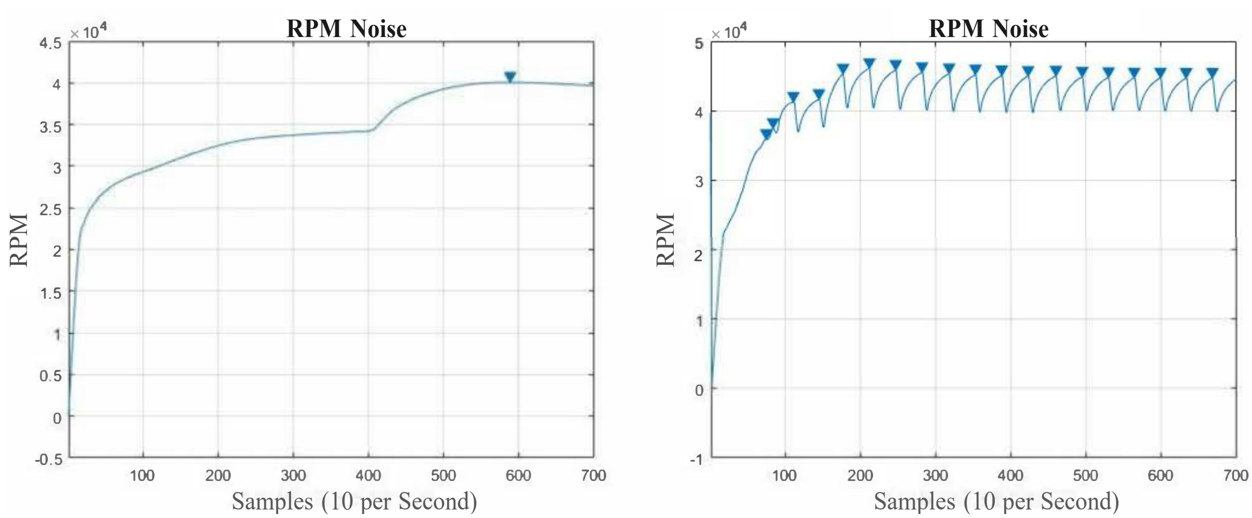

Fig. 13 Smooth and Noisy RPM Responses

An analysis of this data set was done in order to provide an understanding of what the main drivers were for noise in the RPM responses were. By running predictive analysis within the JMP statistical software using neural networks, the variables of importance for producing noise was determined. The variables of importance are shown in descending order in Fig.14 which includes the rpm response since the previous response is used to predict the next time step for the surrogate. Followed by rpm, Tt, Wf36, W, EFTMS, and FCF were the most important inputs. The prediction profiler is shown in Fig.15 which gives the probability of a noisy result (Noisy = 1) on bottom, and probability of non-noisy result (Noisy = 0) on top for a given set of variables. Since the noisy result is a discrete response, all variable slopes are steep. The profiler is for a specific set of variables so when high impact variables are on either end of their range, low impact variables do not affect the results. From this information, the design space of the important variables were reduced in order obtain less noisy RPM data. Tt was reduced from 523 - 1200 R to 560 - 800 R, Wf36 from 0 - 20 pps to 0 - 7 pps, W from 17 - 400 pps to 17 - 300 pps, EFTMS from 79.5-332.25 kW to 79.5-300 kW, and FCF from 1E-4 - 7 pps to 1E-4 - 3 pps.These input ranges were still greater than or equal to the ranges give by an example mission profile. After re-sampling with the new input ranges, the number of noisy response decreased to 26%.

By implementing the noise detection into the sampling procedure, the remaining noisy cases were removed from the rpm training data. This allows for the surrogate to avoid unnecessary error caused by the model noise. An important note here is the fact that rpm is often noisy while the other responses are smooth. For creating surrogate on other outputs, the noisy rpm cases should still be included in the training.

Column	Main Effect	Total Effect
rpm	0.654	0.917
Tt	0.019	0.109
Wf36	0.015	0.098
W	0.023	0.09
EFTMS	0.019	0.09
FCF	0.016	0.071
DEW	0.012	0.05

Fig. 14 Variables Importance

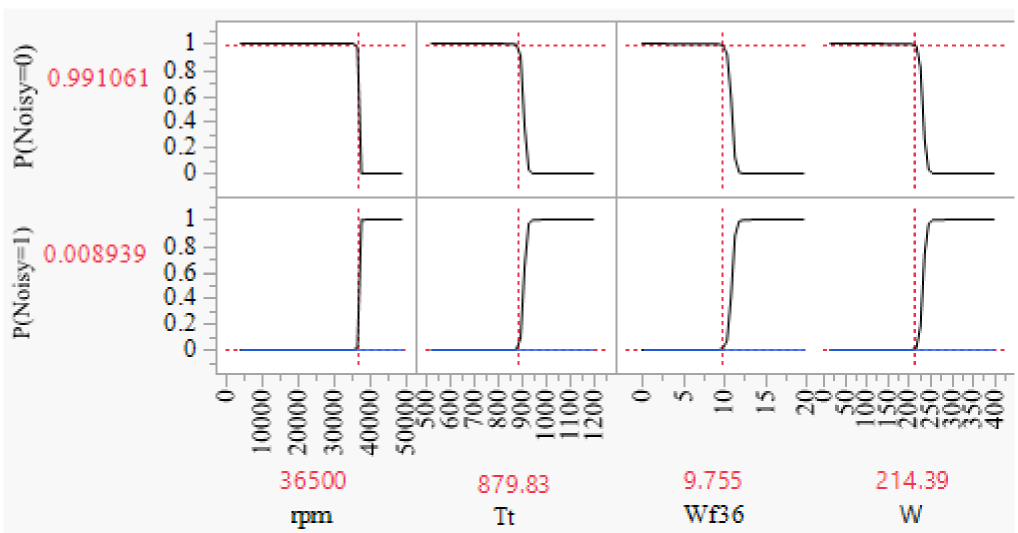

Fig. 15 Predition Profiler

IV. Results

Preliminary results were obtained for the RPM response, which was the nosiest out of all model outputs. These results were tested against the DoE cases where data was collected using a single step function. Therefore, these tests are not indicative of a full mission profile. However, they do give a good first indication of how the surrogate will perform. In Fig.16 the RPM surrogate was tested on training data which was used to fit the surrogate. It shows an up to 1% error at around five seconds.

The full time trace of the sample mission for RPM is shown in Fig. 17. This gives the model data in red and surrogate data in black. The surrogate follows the TMS model fairly closely throughout the time trace. For much of the mission, before 7,000 seconds, the error is less than 5%. The maximum error is around 10%. It can be seen that the maximum error is encountered momentarily at the times of change in the inputs, corresponding to the one the schedule in Fig. 1. The points which deviate from the y=x line in the Actual by Predicted RPM plot and the pattern in the Residual by Predicted RPM plot in Fig.17 are also a result of the error at the times of change of RPM. Overall it shows the surrogate model works but has room for improvement.

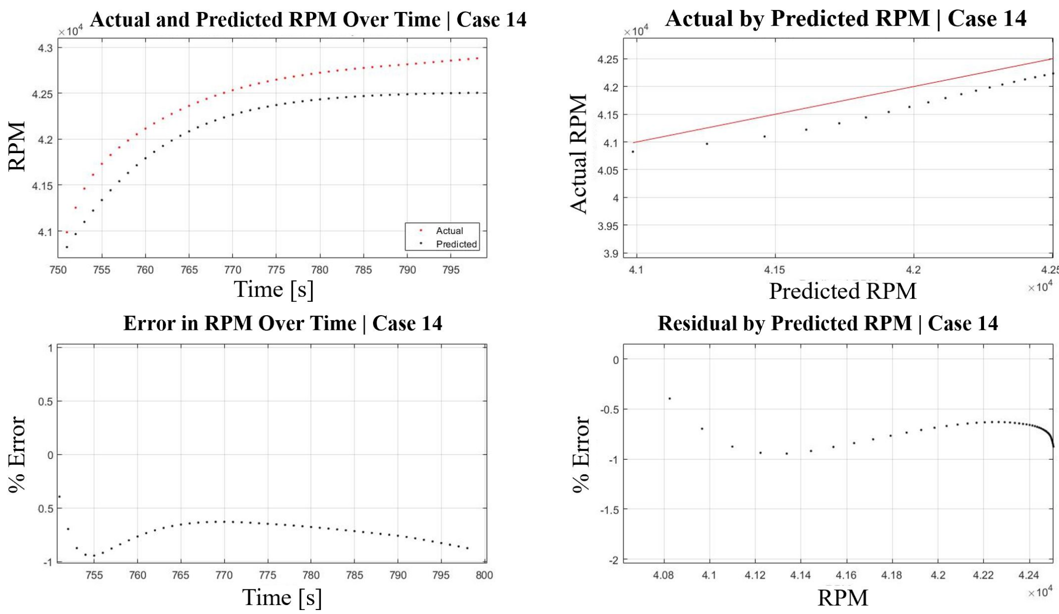

Fig. 16 Initial Test Results for RPM Surrogate

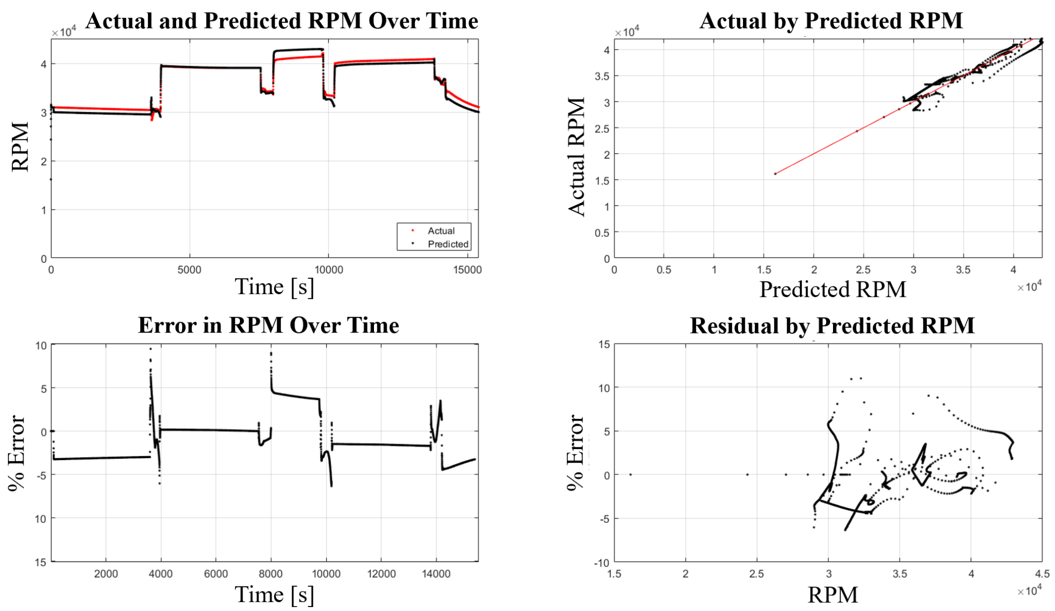

Fig. 17 RPM Full Time Trace Results

V. Conclusion

This paper shows the process for creating time dependent surrogate models for TMS models. The data from the RPM model proves the methodology as viable for creating transient surrogate models. This method is viable for reducing the computational burden from running high fidelity thermal management systems. The TMS model for this project takes approximately 15 minutes to run whereas the surrogate models may be executed within seconds. These surrogates are viable for any set of input schedules within the ranges provided and allow for easy transfer of TMS information to aircraft level studies. By including noisy detection algorithms models which produce high variant results may be re-sampled to avoid such discrepancies. Noisy data may also be removed from sampling of surrogates in order to reduce the inaccuracies caused by training surrogates on inaccurate data. Sampling the model after steady state was reached allows for the highly transient data to be avoided at the start of the model which is mainly caused by model convergence and allows for much less dense sampling rate. To improve upon this, further work could involve creating more cases to gather more sampling data for training of the surrogates as well as modifying the DoE to better account for the correlations input variables.

VI. Acknowledgments

The authors would like acknowledge Dr. Soumya Patnaik and AFRL for providing their resources and insight into transient thermal management systems.

References

[1] Deppen, T., Raczkowski, B., Amrhein, M., Wells, J., Walters, E., Bodie, M., and Patnaik, S., "A Specification Analysis Framework for Aircraft Systems," *SAE International Journal of Aerospace*, Vol. 9, No. 1, 2016, pp. 82–90. doi:10.4271/2016-01-2023.

[2] Yondo, R., Andrés, E., and Valero, E., "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses," *Progress in Aerospace Sciences*, Vol. 96, 2018, pp. 23–61. doi:10.1016/j.paerosci.2017.11.003.

[3] Koziel, S., and Leifsson, L. (eds.), *Surrogate-Based Modeling and Optimization*, Springer New York, 2013. doi:10.1007/978-1-4614-7551-4.

[4] Balchanos, M., Moon, K., Weston, N. R., and Mavris, D. N., "A Dynamic Surrogate Model Technique for Power Systems Modeling and Simulation," *SAE Technical Paper Series*, SAE International, 2008. doi:10.4271/2008-01-2887.

[5] Miller, C., Maser, A., Garcia, E., and Mavris, D., "INVENT Surrogate Modeling and Optimization of Transient Thermal Responses," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-1123.

[6] Hu, Z., and Mahadevan, S., "A Single-Loop Kriging Surrogate Modeling for Time-Dependent Reliability Analysis," *Journal of Mechanical Design*, Vol. 138, No. 6, 2016, p. 061406. doi:10.1115/1.4033428.

[7] Phan, L. L., "A Methodology for the Efficient Integration of Transient Constraints in the Design of Aircraft Dynamic Systems," Ph.D. thesis, Georgia Institute of Technology, 2010.

[8] Rezvani, R., Ozcan, M., Kestner, B., Tai, J., Mavris, D. N., Meisner, R., and Sirica, S., "A Gas Turbine Engine Model of Transient Operation Across the Flight Envelope," *ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition*, ASME, 2011. doi:10.1115/gt2011-45565.

[9] Ozcan, M. F., Kestner, B., Tai, J. C., and Mavris, D. N., "Pattern Classification of a Civilian Turbofan's State Space for Real-Time Surrogate Modeling," *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, American Institute of Aeronautics and Astronautics, 2013. doi:10.2514/6.2013-4003.

[10] Kestner, B. K., Tai, J. C., and Mavris, D. N., "A Computationally Efficient Methodology for Generating Training Data for a Transient Neural Network of a Tip-Jet Reaction Drive System," *Journal of Engineering for Gas Turbines and Power*, Vol. 133, No. 12, 2011, p. 121601. doi:10.1115/1.4003957.

[11] Verghese, G., Levy, B., and Kailath, T., "A generalized state-space for singular systems," *IEEE Transactions on Automatic Control*, Vol. 26, No. 4, 1981, pp. 811–831. doi:10.1109/tac.1981.1102763.

[12] Kania, M., Koeln, J., Alleyne, A., McCarthy, K., Wu, N., and Patnaik, S., "A Dynamic Modeling Toolbox for Air Vehicle Vapor Cycle Systems," *SAE Technical Paper Series*, SAE International, 2012. doi:10.4271/2012-01-2172.

[13] Ponniah, P., *Database Design and Development*, Wiley-Blackwell, 2003. URL https://www.ebook.de/de/product/3606073/paulraj_ponniah_database_design_and_development.html.

[14] Venkateshan, S., "Design of Experiments," *Mechanical Measurements*, John Wiley & Sons, Ltd, 2015, pp. 73–99. doi:10.1002/9781119115571.ch3.

[15] Mavris, D., "Design-of-Experiments For Practical Applications in Modeling, Simulation, and Analysis," AE-6373: Advanced Design Methods 1 course notes, School of Aerospace Engineering, Georgia Institute of Technology, 2018.